Lumos: A Cognitive Control-Inspired Agent for Long-Horizon Threat Investigation

Die Hu[1,2], **Bowei Dai**[3,4,*] **Weitao Tang**[2], **Jingguo Ge**[1,2,*] **He Kong**[1,2], **Hui Li**[1]
[1]State Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Institute of Microelectronics of Chinese Academy of Sciences
[4]School of Integrated Circuits, University of Chinese Academy of Sciences
[*]Corresponding authors

Abstract

Large language model (LLM) agents have shown strong capabilities in long-horizon tasks such as cyber threat analysis. However, despite competent step-level reasoning, existing agents often fail to maintain coherent behavior over extended investigation trajectories, leading to accumulated process-level errors. Prior approaches mainly improve individual reasoning steps through prompting or reflection, but provide limited support for sustained goal maintenance, strategy regulation, and execution monitoring. We propose Lumos, a cognitively inspired agent framework that realizes an explicit executive-control loop via process-level control mechanisms, including goal-centric memory for long-term goal maintenance, failure-driven strategy adaptation for inhibitory regulation, and action and output control for explicit monitoring and verification. Experiments on real-world security investigation scenarios demonstrate that Lumos consistently outperforms strong baselines, improving long-horizon stability and overall task success.

Keywords: Cognitive science; Long-horizon reasoning; LLM agents; Cyber threat investigation; Executive control

Introduction

High-value cognitive tasks are mostly not single-shot decisions, but extended processes that require sustained goal maintenance, evidence integration, and repeated validation over long time horizons (D'Amico & Whitley, 2008; D'Amico et al., 2005). Cyber Threat Investigation (CTI) in Security Operation Centers (SOCs) is a representative example. Analysts must triage large volumes of alerts under uncertainty, linking evidence across multiple log sources to determine whether an attack has occurred and to reconstruct its causal trajectory (Ainslie et al., 2023; Guarascio et al., 2022; Jo et al., 2022). Such investigations often span hours or days and involve iterative querying and hypothesis revision over partially observable environments (Li et al., 2024). Threat investigation closely resembles classic complex problem-solving tasks, characterized by long temporal structure, partial observability, and accumulating process constraints (Funke, 2001). Such tasks constitute a central focus in cognitive science research on executive control and metacognition.

In recent years, large language models (LLMs) have demonstrated strong capabilities in multi-step reasoning and tool use, motivating their adoption as automated investigation agents (Guo et al., 2024; Mohammadi et al., 2025). Existing approaches typically enhance step-level decision quality through techniques such as chain-of-thought prompting (Wei et al., 2022), explicit tool invocation (Yao et al., 2023), or self-reflection mechanisms (Shinn et al., 2023). However, whether such approaches can sustain coherent behavior over extended investigation trajectories remains an open question.

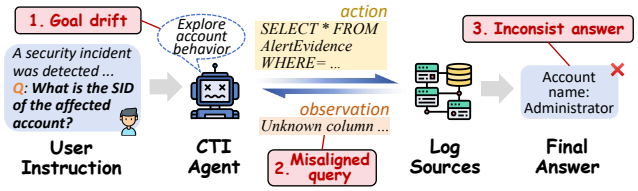

Figure 1: Example failure trajectory with process-level breakdowns at goal, action, and output stages.

Our experiments on real-world SOC scenarios (Wu et al., 2025) show that even when models exhibit strong step-level reasoning ability, current LLM agents consistently display process-level failures in long-horizon investigations (Press et al., 2023). Specifically, as illustrated in Figure 1, we identify three failure patterns that are highly consistent across methods: (1) **Goal Drift**, where the agent gradually deviates from the original investigation objective after multiple rounds of querying and reasoning, shifting toward unconstrained exploration around intermediate cues (Liu et al., 2024); (2) **Query Misalignment**, in which generated SQL queries omit critical constraints (e.g., timestamps or alert identifiers) or become overly generalized and potentially unsafe, violating domain-specific operational rules (Vinay, 2025); (3) **Answer Inconsistency**, where the final output fails to match the required entity type, field, or format—for example, returning hostnames when IP addresses are requested, or mixing residual reasoning text into structured answers (Chung et al., 2025).

From a cognitive science perspective, these failure patterns closely consistent with functional breakdowns in executive control and metacognitive monitoring observed in human complex problem solving. Rather than arising from isolated reasoning errors, such failures accumulate over extended task execution when goals, actions, and intermediate results are not continuously constrained (Miller & Cohen, 2001; Norman & Shallice, 1986b). Accordingly, our focus is not on whether an agent can perform individual reasoning steps, but on whether it can maintain cross-step coherence and stability throughout execution, that is, whether it exhibits cognitive stability in long-horizon tasks.

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society*

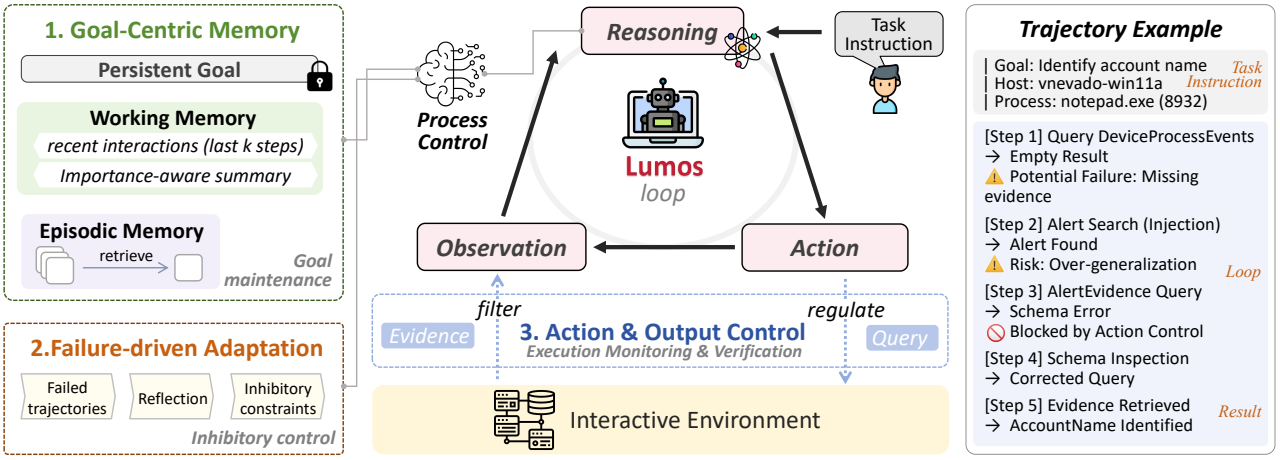

Figure 2: Overall Architecture of **Lumos**. Three process-level control modules regulate a reasoning loop that interacts with the environment and produces the final output.

Building on this analysis, we propose *Lumos*, a cognitive control–inspired LLM agent framework for CTI. Lumos introduces targeted structural control mechanisms into the agent's execution loop to mitigate process-level deviations. To counter goal drift, Lumos maintains long-term goal coherence through explicit goal representations and trajectory management. To prevent violations of domain constraints or omission of critical conditions during tool use, Lumos internalizes security-domain rules as constraints over the action space. In addition, Lumos incorporates a consistency-checking step at the conclusion stage to ensure that final outputs remain aligned with the original task requirements. By embedding these process-level controls at multiple stages, Lumos mitigates the accumulation of errors in long-horizon investigation tasks. Our main contributions are summarized as follows:

- We characterize three stable process-level deviations exhibited by LLM agents in real-world CTI tasks, and systematically map them to cognitive control functions such as goal maintenance, inhibitory control, and output monitoring;
- We propose Lumos, a cognitive control–inspired CTI agent that mitigates error accumulation in long-horizon decision-making through multi-stage process constraints;
- Empirical results on real SOC scenarios show that introducing explicit process-level control mechanisms improves stability in long-horizon investigation tasks.

Problem Formalization

We formalize SOC threat investigation as a long-horizon, goal-directed sequential decision-making problem under partial observability. Formally, the task is modeled as a Partially Observable Markov Decision Process (POMDP) (Kaelbling et al., 1998). Given a natural-language investigation goal g, the agent interacts with a structured log database via a sequence of queries and produces a final conclusion. At each step t, the agent operates on a latent investigation state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ corresponding to a database query or termination, and receives an observation $o_t \in O$ returned by the database (Schick et al., 2023; Yao et al., 2023).

Due to partial observability, the agent can only access a subset of task-relevant information at each step. Task failure therefore rarely stems from an isolated reasoning error. Instead, errors tend to accumulate over time as investigation goals, query conditions, or intermediate conclusions gradually deviate from the original objective (Press et al., 2023).

Methodology

Architecture Overview

We conceptualize long-horizon threat investigation as a problem of cognitive stability rather than isolated reasoning. Effective investigation requires the agent to maintain task goals over time, suppress unproductive strategies, and continuously monitor execution outcomes under partial observability.

Motivated by executive control and metacognitive monitoring (Aron, 2011; Braver, 2012; Miller & Cohen, 2001), Lumos is designed as a unified control framework that regulates the investigation process across multiple timescales. As illustrated in Figure 2, Lumos implements three complementary controls: (1) a goal-centric memory system, (2) failure-driven strategy adaptation, and (3) action and output control.

Goal-Centric Memory System

Long-horizon SOC investigations require agents to integrate evidence across extended interaction sequences under partial observability. Cognitive studies indicate that such behavior critically depends on mechanisms for goal maintenance and capacity-limited memory updating (Baddeley, 2012). Without explicit regulation, task-relevant representations are easily displaced by recent observations, leading to gradual goal drift. To address this challenge, inspired by hierarchical models of human memory, Lumos introduces a hierarchical memory state at each step t:

$$\mathcal{M}_t = \{G, W_t, E\},$$

where G encodes a persistent representation of the investigation goal and its constraints, W_t denotes task-specific working memory, and E stores episodic memory from prior investigations. This separation enables differential update dynamics, providing a stable substrate for long-horizon reasoning.

Goal Anchoring. To counteract recency bias in long-context reasoning, Lumos explicitly maintains the original investigation goal g as a persistent representation G. Unlike transient contextual content, G is not subject to routine context updates or truncation. Instead, it functions as a top-down control signal that continuously biases reasoning toward the original investigative objective, stabilizing goal-directed behavior across extended interaction horizons.

Working Memory Regulation. As investigation progresses, the Thought-Action-Observation trajectory $\tau_{1:t}$ grows rapidly. Naively accumulating the full interaction history leads to context overload and introduces large amounts of decision-irrelevant noise, degrading reasoning stability and goal alignment. To address this, reflecting the limited capacity of human working memory under cognitive load, Lumos introduces a novel working memory as shown in Figure 3.

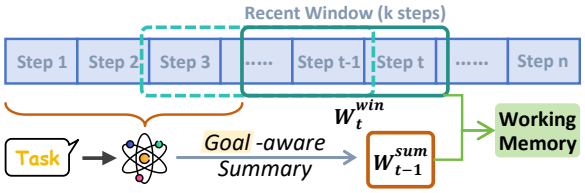

Figure 3: Goal-aware working memory regulation in Lumos.

Rather than retaining the full interaction history, Lumos regulates working memory through an asymmetric update strategy. At time step t, the agent's working memory is partitioned as $W_t = \{W_t^{win}, W_{t-1}^{sum}\}$, where W_t^{win} retains the most recent k interaction steps in full detail, and W_{t-1}^{sum} is a compressed representation of earlier trajectory segments that preserves long-range investigative continuity. When the trajectory length exceeds a predefined compression threshold T, Lumos applies an importance-aware compression process conditioned on the persistent goal G:

$$S_t = f_{\text{sum}}(\tau_{1:t-k}).$$

The summarized memory W_{t-1}^{sum} is constrained to a maximum length L to prevent summary overgrowth. This summarization prioritizes information that remains causally or semantically relevant to the investigation objective, such as key entities, pivotal queries, and major reasoning transitions. By constraining memory growth while retaining decision-critical content, this mechanism reduces contextual noise and mitigates long-horizon drift.

Episodic Memory Retrieval. In human cognition, episodic memory supports problem solving not through continuous activation, but through selective retrieval when current strategies fail. Consistent with this view, Lumos maintains an episodic memory store

$$E = \{(\tau^{(i)}, y^{(i)})\},$$

where each $\tau^{(i)}$ corresponds to a completed historical investigation trajectory and $y^{(i)} \in \{0, 1\}$ indicates whether the investigation satisfied the task goal.

At the beginning of a new investigation, episodic memory is selectively retrieved to support strategy selection as shown in Figure 4. Given the current query representation q, Lumos retrieves a small set of similar past trajectories:

$$E_q = \{\tau^{(i)} \mid (\tau^{(i)}, y^{(i)}) \in E,\ \text{sim}(\phi(q), \phi(\tau^{(i)})) \in \text{Top-}k\},$$

where $\phi(\cdot)$ encodes the query and trajectory summaries, and E_q denotes the top-k episodic trajectories retrieved by cosine similarity.

Importantly, retrieved episodic traces do not provide factual answers directly. Instead, they are injected into the current context as procedural priors, influencing subsequent query expansion strategies and constraint completion. Through this analogy-based reuse of experience, Lumos reduces unproductive exploration within a constrained action space, thereby accelerating convergence in complex investigations.

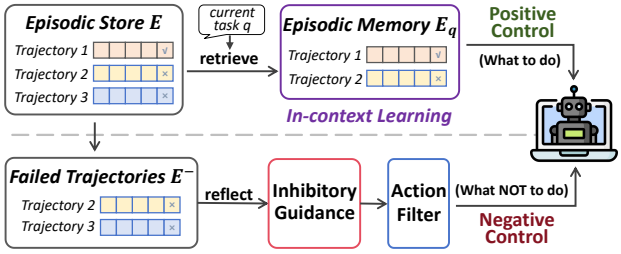

Figure 4: Episodic memory in Lumos supports both episodic retrieval and failure-driven reflection.

Failure-Driven Strategy Adaptation.

In long-horizon scenarios, task failure often arises not from a lack of knowledge, but from perseverative use of ineffective strategies. Cognitive studies suggest that adaptive problem solving relies on inhibitory control mechanisms that suppress unproductive behaviors following failure (Flavell, 1979; Nelson, 1990).

Motivated by this view, Lumos introduces a failure-driven strategy adaptation mechanism that regulates strategy selection through negative experience, as shown in Figure 4. Unlike episodic memory, which supports experience reuse via retrieval, negative experience in Lumos is distilled into inhibitory constraints that specify which reasoning or action patterns should be avoided.

Failure Attribution via Reflection When an investigation exhibits clear failure or stagnation, Lumos triggers a reflection phase. During reflection, a subset of failed trajectories is sampled from episodic memory, $E^- = \{\tau \in E \mid y = 0\}$, and

processed by a reflection function $f_{\text{ref}}(\cdot)$ to produce a set of inhibitory constraints:

$$C_t^- = f_{\text{ref}}(E^-).$$

Rather than replaying concrete queries or log contents, C_t^- captures abstract failure attributions, such as query patterns that frequently cause execution errors, invalid hypothesis forms, or reasoning trajectories that diverge from the investigation goal. In our implementation, f_{ref} is a lightweight LLM-based abstraction module invoked only upon failure signals. An illustrative reflection prompt is simplified shown below:

> **Reflection Prompt.** You are given summaries of failed investigation attempts on long-horizon security analysis tasks. Analyze these failures and identify recurring strategy-level patterns that should be avoided in future attempts. Focus on high-level reasoning or action patterns rather than specific queries or answers. Do not propose new actions or solutions. Instead, produce 2–3 inhibitory constraints, each describing a class of behaviors that led to failure. Each constraint should be stated in the form: "Avoid ... when ...".

Inhibitory Guidance Injection and Strategy Regulation From a cognitive perspective, this mechanism corresponds to inhibitory control over perseverative behavior, a hallmark of executive dysfunction observed in clinical populations.

The induced inhibitory constraints C_t^- are injected into the agent's reasoning context as negative guidance. Conceptually, this contracts the feasible action space at time step t:

$$\mathcal{A}_t^{\text{valid}} = \mathcal{A}_t \setminus C_t^-.$$

In practice, this contraction is implemented by incorporating C_t^- as negative constraints during action generation, suppressing candidates that match known failure patterns. This 'offline reflection–online inhibition' mechanism progressively reduces unproductive strategy reuse without parameter updates, guiding Lumos toward more reliable behavior.

In summary, failure-driven strategy adaptation equips Lumos with an explicit inhibitory control mechanism. While memory preserves what the agent knows, inhibitory regulation governs what the agent learns to stop doing, jointly supporting stable long-horizon reasoning.

Action and Output Control

Where previous modules regulate internal cognition, this module constrains the agent's interaction with the environment. Inspired by cognitive theories of executive control, Lumos introduces explicit control mechanisms over action execution, observation intake, and output verification (Figure 5), enforcing structured constraints at multiple execution stages. From a cognitive perspective, these mechanisms support long-horizon agent regulation by combining proactive control (goal maintenance and pre-action constraint enforcement) with reactive adjustment (failure-triggered reflection). We distinguish normative validity constraints (e.g., syntax/schema checks required by the environment) from goal-conditioned control constraints (e.g., policy gating and output verification). Our cognitive interpretation primarily concerns the latter.

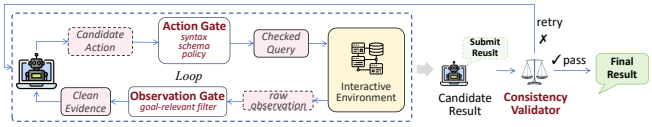

Figure 5: Action and output control in Lumos.

Constraint-Gated Action Execution At step t, the agent generates a candidate action $\tilde{a}_t$ conditioned on the current memory state $\mathcal{M}_t$. In Lumos, action generation is explicitly decoupled from action execution: a candidate action is executed only after passing a constraint-aware gating process. We formalize action validity using the following indicator function:

$$\mathbb{I}(\tilde{a}_t) = \mathbb{I}_{\text{syntax}}(\tilde{a}_t) \wedge \mathbb{I}_{\text{schema}}(\tilde{a}_t) \wedge \mathbb{I}_{\text{policy}}(\tilde{a}_t \mid G),$$

where $\mathbb{I}_{\text{syntax}}$ verifies SQL syntactic correctness, $\mathbb{I}_{\text{schema}}$ enforces consistency with database schemas and field types, and $\mathbb{I}_{\text{policy}}$ applies domain-specific constraints aligned with the investigation goal G and security policies (e.g., read-only access or temporal window restrictions).

A candidate action is submitted to the environment as an executed action a_t only if $\mathbb{I}(\tilde{a}_t) = 1$. Otherwise, execution is blocked and the invalid action is rejected before interacting with the environment. In our implementation, syntactic and schema constraints are enforced via deterministic SQL parsers and database drivers, while policy constraints are encoded as rule-based checks conditioned on G. This execution gating localizes potential execution errors to a single step, preventing invalid queries from contaminating subsequent observations and memory updates.

Observation Filtering Query execution often produces observations that contain substantial information unrelated to the current investigation objective. Inspired by theories of selective attention (Norman & Shallice, 1986a), Lumos applies sensory-level filtering to regulate the flow of environmental feedback in a goal-conditioned manner.

Formally, the raw observation $\tilde{o}_t$ is transformed by a gating function $g_{\text{gate}}(\cdot)$ into a controlled observation:

$$o_t = g_{\text{gate}}(\tilde{o}_t),$$

which emphasizes only records and attributes causally relevant to the current goal G and recent reasoning context. By filtering irrelevant or redundant information before integration into working memory W_t, this mechanism reduces observation-level noise and complements the trajectory compression strategy introduced before.

Output Consistency Verification Before terminating an investigation episode, Lumos introduces an explicit metacognitive-style verification step that evaluates the candidate output $\hat{y}$ as an object of inspection rather than as the direct endpoint of reasoning. We formalize this verification using a decision function:

$$J(\hat{y}, g) \in \{0, 1\},$$

Table 1: Performance comparison across all incidents (Success Rate ↑ / Average Reward ↑).

Type	Model	Incident 5		Incident 34		Incident 38		Incident 39		Incident 55		Incident 134		Incident 166		Incident 322		Overall	
		Succ.	Rew.	Succ.	Rew.	Succ.	Rew.	Succ.	Rew.	Succ.	Rew.	Succ.	Rew.	Succ.	Rew.	Succ.	Rew.	Succ.	Rew.
Base Model	gpt-4o	36.7	0.338	29.3	0.293	36.4	0.364	29.6	0.273	29.0	0.249	49.1	0.491	17.2	0.166	32.1	0.315	31.1	0.293
	o1-mini	14.3	0.147	24.4	0.244	9.1	0.091	22.4	0.230	15.0	0.160	33.3	0.333	18.4	0.189	37.5	0.382	21.7	0.222
	gemini2.5 flash	30.6	0.312	30.5	0.329	36.4	0.364	24.5	0.248	21.0	0.224	49.1	0.491	24.1	0.260	37.5	0.375	29.5	0.305
	qwen-3-32b	18.4	0.191	22.0	0.229	9.1	0.091	20.4	0.207	11.0	0.116	19.3	0.200	11.5	0.133	23.2	0.250	17.3	0.182
	gork-4	29.6	0.306	41.5	0.424	45.5	0.455	30.6	0.320	24.0	0.248	50.9	0.509	27.6	0.280	41.1	0.421	33.6	0.344
	Claude-haiku-4	22.4	0.224	24.4	0.263	36.4	0.364	22.4	0.236	13.0	0.136	24.6	0.253	12.6	0.140	14.3	0.161	17.8	0.204
	deepseek-chat	29.6	0.306	33.3	0.333	36.4	0.364	13.3	0.147	7.0	0.092	14.0	0.147	5.7	0.062	3.6	0.057	13.5	0.148
Agent (4o-mini)	React	26.5	0.278	26.8	0.268	54.5	0.545	16.3	0.165	24.0	0.244	33.3	0.333	28.7	0.287	37.5	0.382	30.9	0.313
	Reflexion	40.8	0.408	50.0	0.505	54.5	0.545	36.7	0.367	47.0	0.480	52.6	0.526	41.4	0.414	50.0	0.511	46.6	0.470
	Expel	36.7	0.371	34.1	0.341	18.2	0.182	28.6	0.293	26.0	0.260	29.8	0.298	26.4	0.269	37.5	0.382	29.7	0.299
	Lumos (Ours)	**55.1**	**0.559**	**62.2**	**0.629**	**73.3**	**0.742**	**45.9**	**0.464**	**51.0**	**0.516**	**74.6**	**0.751**	**58.6**	**0.592**	**56.8**	**0.575**	**61.2**	**0.620**
Agent (DeepSeek)	React	26.5	0.284	37.8	0.388	18.2	0.182	16.3	0.165	10.0	0.116	14.0	0.154	5.7	0.057	26.8	0.271	19.2	0.201
	Reflexion	13.3	0.144	12.2	0.127	9.1	0.127	12.2	0.133	17.0	0.186	14.0	0.154	9.2	0.094	16.1	0.169	13.4	0.144
	Expel	19.4	0.218	30.5	0.310	27.3	0.309	21.4	0.242	14.0	0.187	22.8	0.235	10.3	0.108	19.6	0.232	19.5	0.218
	Lumos (Ours)	**51.0**	**0.520**	**61.0**	**0.615**	**72.7**	**0.764**	**42.9**	**0.433**	**48.0**	**0.490**	**71.9**	**0.719**	**57.5**	**0.579**	**55.4**	**0.571**	**54.3**	**0.551**

which assesses output validity along three dimensions: *Type Consistency*, ensuring the answer entity matches the task requirement; *Format Consistency*, verifying adherence to predefined output specifications; and *Semantic Alignment*, checking whether the output directly addresses the investigation goal g rather than reflecting intermediate reasoning artifacts. This verification step is not intended to model internal metacognitive reasoning processes, but to operationalize the functional role of metacognition in ensuring goal-consistent task completion.

If $J(\hat{y}, g) = 1$, the answer is accepted and submitted. Otherwise, termination is blocked and the verification failure is injected into working memory as structured feedback, triggering a controlled regeneration process. This self-evaluation mechanism mirrors metacognitive checks commonly applied by human investigators prior to task completion.

Together, these execution-level controls regulate how strategies are instantiated as concrete actions and finalized outputs. By enforcing constraints at action execution, observation intake, and response submission, Lumos prevents local execution errors from propagating across long decision trajectories.

Experiments

Experimental Setup

Dataset. We evaluate Lumos on a security incident investigation benchmark consisting of 589 questions spanning 8 real-world security incidents (Wu et al., 2025). Each incident corresponds to a distinct attack scenario (e.g., network intrusion, malware execution, lateral movement) and requires multi-step reasoning over structured security logs, including timestamps, IP addresses, process identifiers, and alert records. Answering each question typically involves iterative database querying, evidence aggregation, and causal reasoning, making the benchmark well suited for evaluating long-horizon investigative agents.

Metrics. We employ below metrics to evaluate CTI agent: (1) **Success Rate**: The percentage of questions where the agent provides a complete and correct answer (binary); (2) **Average Reward**: A continuous score accounting for partial credit in multi-step investigations (ranging from 0 to 1).

Compared Methods. We compare Lumos against four representative baselines: (1) **Baseline**, which directly prompts the language model without an agent framework; (2) **ReAct** (Yao et al., 2023), a standard thought-action-observation reasoning loop; (3) **Reflexion** (Shinn et al., 2023), which augments ReAct with self-reflection and error correction; and (4) **Expel** (Zhao et al., 2024), an experience-pool-based method with retrieval-augmented generation.

Implementation Details. All experiments are conducted using DeepSeek-V3.2 and 4o-mini as backbone models, with temperature set to 0 and a maximum of 15 reasoning steps per question to ensure deterministic behavior. Fixed cache seeds are used for all methods to ensure consistent database query results and fair token comparison. Lumos activates reflection learning, SQL validation, trajectory management with a compression threshold of 18, preservation of the eight most recent messages, summaries capped at 800 characters, and importance-aware extraction, as well as a judger with a single retry. The retrieval-augmented generation module utilizes an Experience Pool embedding both question and context, with ‘sentence-transformers/all-mpnet-base-v2 embeddings‘.

Performance Comparison

We analyze the empirical performance of Lumos in comparison with representative baselines, focusing on success rate and average reward across different incident scenarios.

As reported in Table 1, Lumos consistently outperforms all baselines across both backbones. On the DeepSeek-Chat backbone, Lumos achieves a 54.3% success rate, representing a 40.8% absolute gain over the base model (13.5%) and the Reflexion agent (13.4%). This consistent gap across heterogeneous architectures suggests that Lumos provides a systematic performance lift that is not tightly coupled to specific backbone capabilities.

Also, Lumos maintains substantially more stable execution as investigation depth increases. On incident 39, while ReAct and Reflexion degrade to ∼15% success, Lumos sustains over 42%; a similar pattern appears on incident 38, where Lumos exceeds ReAct by 54.5 points. These results indicate stronger resistance to error accumulation, enabling Lumos to

preserve valid investigative trajectories where baseline agents frequently derail. This stability is mirrored by aligned gains in average reward. For incident 34, Lumos (4o-mini) achieves a reward of 0.629, more than doubling ReAct (0.268), indicating consistent multi-step evidence aggregation rather than isolated correct guesses.

Besides, Lumos substantially outperforms the adaptation-based agent Expel on the 4o-mini backbone, achieving a success rate of 61.2% compared to Expel's 29.7%. This gap suggests that structured execution constraints are more effective for the security domain than purely episodic memory retrieval, as they provide more consistent execution-level guidance during complex multi-step investigations.

Ablation Study

To verify the effectiveness of each module in Lumos, we conduct ablation experiments on the DeepSeek-Chat backbone, as shown in Table 2.

Table 2: Ablation study of Lumos control systems.

Configuration	Success Rate (%)	Avg. Reward
Lumos (Full Model)	**54.3**	**0.551**
w/o MS	48.2	0.485
w/o SA	50.5	0.512
w/o AO	45.4	0.461

Removing the Memory System Control (w/o MS) results in a 6.1 percentage point decrease in success rate (48.2%). This indicates that without trajectory compression and goal anchoring, reasoning is more vulnerable to context overload and long-horizon goal drift.

Restricting the Strategy Adaptation Control (w/o SA) leads to a 3.8 percentage point reduction, reflecting the value of failure-driven reflection. This suggests that failure-driven adaptation helps suppress repeated ineffective exploration through inhibitory constraints derived from negative experience.

Disabling the Action and Output Control module (w/o AO) leads to the most significant performance degradation, with the success rate dropping by 8.9 percentage points. This confirms that without constraint-aware gating and output verification, agents more frequently suffer from execution-level failures such as malformed queries and inconsistent outputs.

These ablations also align with the three process-level failure patterns identified earlier: memory control mitigates goal drift, strategy adaptation suppresses repeated ineffective exploration, and action/output control reduces query misalignment and answer inconsistency. Taken together, the results suggest that Lumos' gains arise from complementary process-level regulation rather than any single reasoning component.

Hyperparameter Sensitivity Analysis

Figure 6 presents the sensitivity of Lumos to key memory-related hyperparameters. Performance exhibits a clear unimodal trend, peaking at a compression threshold of T=18, with k=8 recent messages and a summary length of L=800 tokens, achieving a success rate of 54.3%. Deviations from these values lead to substantial performance degradation (10-20% absolute). Smaller thresholds or summaries result in premature information loss, while larger values introduce excessive context that degrades decision quality. These results indicate that effective long-horizon investigation relies on calibrated memory regulation rather than maximal context retention.

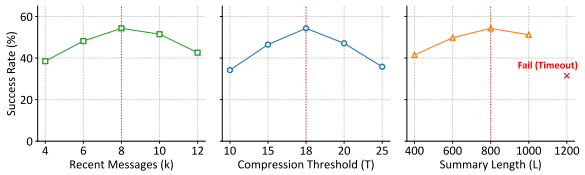

Figure 6: Sensitivity of Lumos to key hyperparameters

Efficiency Analysis

Table 3 reports the efficiency comparison in terms of reasoning steps, database queries, token consumption(billable tokens), latency, and success rate.

Table 3: Efficiency comparison across investigation methods.

Method	Steps ↓	Queries ↓	Tokens ↓	Latency (s) ↓	Success (%) ↑
Baseline	10.8	9.9	1,585	39.5	31.1
ReAct	10.4	9.9	1,539	38.6	19.2
Reflexion	11.0	10.4	86,138	40.8	13.4
Expel	6.8	6.1	173,023	25.0	19.5
Lumos	**7.1**	**6.2**	**117,013**	**26.1**	**54.3**

Lumos achieves the highest success rate among all methods, reaching 54.3%, while requiring only 7.1 reasoning steps and 6.2 database queries on average, indicating a more focused investigation process. Although Lumos incurs higher per-step token usage due to explicit process-level control (e.g., memory regulation and action validation), these mechanisms reduce failed queries and backtracking. Compared with Expel, Lumos attains a substantially higher success rate with lower token consumption, suggesting that structured execution control is more efficient than brute-force context expansion. Lumos also maintains comparable latency at 26.1 seconds.

Conclusion

In this work, we present Lumos, a cognitively inspired agent framework for long-horizon security investigation. On a real-world benchmark, Lumos outperforms all baselines. Empirical analyses show that this improvement arises from coordinated control over memory, action execution, and output validation. Such results highlight the importance of process-level regulation for preventing task deviation in multi-step reasoning. Future work will investigate reinforcement learning–based control policies to optimize memory management and action selection through interaction-driven feedback.

Acknowledgments

This work was supported by Grant No. E4GZ08020403. We also thank the Artificial Intelligence Chip and Systems R&D Center, Institute of Microelectronics of the Chinese Academy of Sciences, for their support.

References

Ainslie, S., Thompson, D., Maynard, S., & Ahmad, A. (2023). Cyber-threat intelligence for security decision-making: A review and research agenda for practice. *Computers & Security*, *132*, 103352.

Aron, A. R. (2011). From reactive to proactive and selective control: Developing a richer model for stopping inappropriate responses [Prefrontal Cortical Circuits Regulating Attention, Behavior and Emotion]. *Biological Psychiatry*, *69*(12), e55–e68. https://doi.org/https://doi.org/10.1016/j.biopsych.2010.07.024

Baddeley, A. (2012). Working memory: Theories, models, and controversies. *Annual review of psychology*, *63*, 1–29. https://doi.org/10.1146/annurev-psych-120710-100422

Braver, T. S. (2012). The variable nature of cognitive control: A dual mechanisms framework. *Trends in cognitive sciences*, *16 2*, 106–13. https://api.semanticscholar.org/CorpusID:35068

Chung, A., Zhang, Y., Lin, K., Rawal, A., Gao, Q., & Chai, J. (2025). Evaluating long-context reasoning in llm-based webagents. https://arxiv.org/abs/2512.04307

D'Amico, A., & Whitley, K. (2008). The real work of computer network defense analysts: The analysis roles and processes that transform network data into security situation awareness. *VizSEC 2007: Proceedings of the workshop on visualization for computer security*, 19–37.

D'Amico, A., Whitley, K., Tesone, D., O'Brien, B., & Roth, E. (2005). Achieving cyber defense situational awareness: A cognitive task analysis of information assurance analysts. *Proceedings of the human factors and ergonomics society annual meeting*, *49*(3), 229–233.

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, *34*(10), 906.

Funke, J. (2001). Dynamic systems as tools for analysing human judgement. *Thinking & reasoning*, *7*(1), 69–89.

Guarascio, M., Cassavia, N., Pisani, F. S., & Manco, G. (2022). Boosting cyber-threat intelligence via collaborative intrusion detection. *Future Generation Computer Systems*, *135*, 30–43.

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., & Zhang, X. (2024). Large language model based multi-agents: A survey of progress and challenges [Survey Track]. In K. Larson (Ed.), *Proceedings of the thirty-third international joint conference on artificial intelligence, IJCAI-24* (pp. 8048–8057). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2024/890

Jo, H., Lee, Y., & Shin, S. (2022). Vulcan: Automatic extraction and analysis of cyber threat intelligence from unstructured text. *Computers & Security*, *120*, 102763.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1), 99–134. https://doi.org/https://doi.org/10.1016/S0004-3702(98)00023-X

Li, L., Huang, C., & Chen, J. (2024). Automated discovery and mapping att&ck tactics and techniques for unstructured cyber threat intelligence. *Computers & Security*, *140*, 103815.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, *12*, 157–173.

Miller, E., & Cohen, J. (2001). An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, *24*, 167–202. https://doi.org/10.1146/annurev.neuro.24.1.167

Mohammadi, M., Li, Y., Lo, J., & Yip, W. (2025). Evaluation and benchmarking of llm agents: A survey. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 6129–6139.

Nelson, T. O. (1990). Metamemory: A theoretical framework and new findings. In *Psychology of learning and motivation* (pp. 125–173, Vol. 26). Elsevier.

Norman, D. A., & Shallice, T. (1986a). Attention to action. In R. J. Davidson, G. E. Schwartz, & D. Shapiro (Eds.), *Consciousness and self-regulation: Advances in research and theory volume 4* (pp. 1–18). Springer US. https://doi.org/10.1007/978-1-4757-0629-1_1

Norman, D. A., & Shallice, T. (1986b). Attention to action: Willed and automatic control of behavior. In *Consciousness and self-regulation: Advances in research and theory volume 4* (pp. 1–18). Springer.

Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., & Lewis, M. (2023). Measuring and narrowing the compositionality gap in language models. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 5687–5711.

Schick, T., Dwivedi-Yu, J., Dessí, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, *36*, 8634–8652.

Vinay, V. (2025). Failure modes in llm systems: A system-level taxonomy for reliable ai applications. *arXiv preprint arXiv:2511.19933*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824–24837.

Wu, Y., Velazco, M., Zhao, A., Luján, M. R. M., Movva, S., Roy, Y. K., Nguyen, Q., Rodriguez, R., Wu, Q., Albada, M., et al. (2025). Excytin-bench: Evaluating llm agents on cyber threat investigation. *arXiv preprint arXiv:2507.14201*.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations (ICLR)*.

Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y.-J., & Huang, G. (2024). Expel: Llm agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(17), 19632–19642.